1. **Problem formulation:**

The objective of this bachelor’s thesis is to design and develop a system for Natural language control for a Universal robot arm. The system should be capable of translating natural language into executable robot commands in the form of URScripts.

The goals for the thesis are: not goals, problems

* Implementing a neural network which is capable of doing natural language processing
* Designing a parser for translating neural network output to robot executable code.
* Designing a constricted set of robot movements which the universal robot should be capable of

The project will be evaluated based on:

* I don’t know ☹

1. **Introduction:**

An introduction that starts with something interesting

*Interesting start up subjects:*

* 4. Industrial revolution
* Industrial Human robot collaboration (and how it currently sucks)
* Public human robot collaboration? Might suck.
* Start with transformers development in 2017, and go into robot implementation of transformers.
* Automatic Translation of Spanish Natural Language Commands to Control Robot Comands Based on LSTM Neural Network:

Using an interesting start subject/subjects, it enables a transversion into my bachelor product.

*Translation/bridge subject into my bachelor project*

* Transformers implemented by robotics, and its effect on robot NLP collaboration.
* Directly go into the task which I’m solving, being able to talk to robots
* Talk about the overall problem that I am solving, and what solving the problem would mean

The meat of the introduction. We are finished warming up the reader, and now the important information must be given. But what is important information?

*Meat on the bone – Important information*

* Dig into how the problem will be solved in this Bachelor thesis (NLP processing using a neural network, Universal robot kinematics and theory, Parsing from NLP to URscripts)
* Dig into the historical aspect of how transformers has been implemented into robotics, and make a perspective view on how this bachelor thesis is a tribute/further development of robot implementation for Natural language processing (NLP) systems

1. **SOTA:**

* [**Reshaping Robot Trajectories Using Natural Language Commands: A Study of Multi-Modal Data Alignment Using Transformers | IEEE Conference Publication | IEEE Xplore (sdu.dk)**](https://ieeexplore-ieee-org.proxy1-bib.sdu.dk/document/9981810)

This work uses transformers along with a CLIP neural network, to combine text understanding with image meaning, to maneuver a robot in between objects of different fragility properties.

It uses a Franke emika panda robot arm as its manipulator as is therefore very closely related to the same work I do.

I can’t seem to understand the specific function (output) of the BERT neural network, as the output is a changed directory movement such to avoid obstacles. This does not seem as an intuitive use of a transformer model

* [Automatic Translation of Spanish Natural Language Commands to Control Robot Comands Based on LSTM Neural Network | IEEE Conference Publication | IEEE Xplore (sdu.dk)](https://ieeexplore-ieee-org.proxy1-bib.sdu.dk/document/8675641)

Uses a RNN encoder decoder structure to translate natural language into an intermediate action language called Robot Control Language (RCL), this has the purpose of translating the language into a non-interpretable text, which is then used for a mobile robot.

The method of using a transformer to create better representations (less interpretable text) before trying to parse them into robot commands, is also a technique which is made use of in my thesis.

Another difference is the use of LSTM’s instead of my proposed model of choice, which is a transformer model. The prediction will be made, that the pretrained and finetuned models used in my thesis, will be superior to the RNN based models, as the transformers are greater at representing text.

The model was trained by using seven different categorizable actions which all sentences would fall into, then base sentences were made for each of the actions, which served as a data tag and data set respectively. In this thesis, the same idea is used, though for this thesis, it is possible to create combinations of different actions in a single sentence to create more complex movement.

[Development of Indonesian Speech Recognition with Deep Neural Network for Robotic Command | IEEE Conference Publication | IEEE Xplore (sdu.dk)](https://ieeexplore-ieee-org.proxy1-bib.sdu.dk/document/8937275)

Using a regular Deep neural network (feed forward), the authors could create a natural language processing model, which was capable of categorizing words directly into one of six different actions. The final model achieved a success fail ratio of 89.57% based on 10.521 datasets.in which the testing set consisted of 10% of the data, meaning, 1052 datasets.

The paper focuses more on the audible part of the process of natural language processing, making it slightly different from the focus I want to put on my thesis. Though their results using a DNN is very interesting, as the success fail ratio clearly states, that the DNN is capable of being a good natural language processing model, given that the actions are simplistic (only six of them)

[Commanding mobile robot movement based on natural language processing with RNN encoder­decoder | IEEE Conference Publication | IEEE Xplore (sdu.dk)](https://ieeexplore-ieee-org.proxy1-bib.sdu.dk/document/8391185)

Based on an overview, what they have done, is that they have split the task into many smaller tasks, so that the RNN decoder has less work to train.

First, they have a pre-processing step which has the purpose of splitting the sentence, such that each part only has one robot command action with its parameters. Furthermore, it also serves to filter out non important text. Very cool, is possibly applicable and extremely practical in my thesis.

Secondly, they use an RNN encoder decoder model to classify each of the sentences by their probability of being one of the actions.

Last of all there is a post processing function that maps the output from the RNN into actual robot code.

The interesting part of this paper is that it uses a pre-processing part, to make sure that the input for the RNN is only considered to be a single language command. Furthermore, the pre-processing serves to filter out uninteresting words which has no meaning for the text.

For the data, they use their own, by making volunteers write down their preferred natural language command given a picture of a robot and a direction. This sums up to 1600 unique datasets, in which they achieve an accuracy of 79.23% and 73.65 % on sentences with multiple commands.

1. **Segment 1 Neural networks**
   1. **NLP introduction**

*Text introducing NLP with respect to robot commands and how natural language can be ambiguous.*

* What is possible with NLP, what is not.
* Some robot command examples that shows ambiguity.
* General NLP tasks often used.

*Specify the problem needed to be solved, in regard to what kind of sentences needs to be analysed and how. (answer is, we need text analysis methods which is capable of extracting meaning from input)*

* Which general language understanding abilities are needed in my thesis.
* Which NLP abilities should I apply to achieve the general language understanding abilities mentioned before.
  1. **NLP using NN**

*Maybe some introduction to how basic neural networks work?*

* Nodes and biases
* Weights
* Sigmoid functions
* Back propagation
* Training (training, validation, test sets, overfitting, regularisation techniques)

*Maybe some math describing what neural networks are (a large function estimating a model based on large amounts of input and training data)*

* Discriminative modelling (absolutely not sure how this works or what it means, but I am game to figure it out, if it means extra style points for the thesis)

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

*Introduction to the use of neural networks for NLP task*

* How text is transformed into numeric values
* How numeric values representing text is pipelined through a neural network.
* how training such pipeline will make a neural network better at understanding text (numeric representation of text)

*Introduce the concept of encoder decoder models.*

* What kind of design works best for NLP? (encoder-decoder models)
* What is achieved by using encoder decoder models (abilities are split into two categories. Understanding and answering)
* What does the encoder generally do
* What does the decoder generally do
* Summarize by making the connection between encoder/decoder properties and NLP tasks of different natures.
  1. **Token classification**

*Introduce how encoder decoder models are used for different tasks for NLP.*

* Encoder -> text analysis (BERT model types)
* decoder -> text generation (GPT, and GPT2)
* Encoder-decoder -> text generation given complex input “translation, summarization” (BART models)

*Go into the fact that an encoder-based model is the preference and focus for this thesis, due to the task which needs to be done.*

* Introduce the BERT encoder model as the primary model type which will be used for this thesis
* Introduce what the BERT model is, and how it came to be, by introducing what transformers are.
  1. **Transformers**

*Introduction to what transformers is and how they work.*

* The basic idea of what a transformer architecture is (attention based neural network, multiheaded attention layers)
* How it’s used in the encoder decoder design.

*“Include model of transformer graph”*

*Diagram

Description automatically generated*

Figure 1 - probably something akin of this

* Explain overall what each part of the graph does.

*Explain in detail based on the graph how each part works.*

* Go into detail what each part is.
* Multi headed attention (single headed attention also)
* Residuals
* Encoder layers
* Decoder layers
* Input embeddings
* Positional encodings
* Input and output

*Explain the properties that makes transformers a great design based on the alternative use of recurrent neural networks.*

* *Parallelization.*
* *Self-attention Big-O complexity.*
* *GLUE and SUPER GLUE scores at the papers time of release.*

*Explain the need of large training sets and how this problem is overcome using pretrained models like bert (transition into BERT.)*

* Explain how much time would be used to train a neural network.
* Explain very short about the carbon footprint of training a neural network from scratch.
* Introduce concept of pretraining and finetuning
  1. **BERT (Bi-directional – encoder representations transformer)**

*Introduce the concept of BERT (an encoder-based model type)*

* Introduce BERT as a pretrained transformer for finetuning.
* Introduce the effectiveness of the BERT model given its bi-directional training property

*Explain what the bi-directional property of the pretrained model is.*

* Masked word learning technique for word contextualization learning.
* Maybe short about sentence contextualization learning methods which is also used in the pretraining step of the BERT model.

*Explain the fine-tuning process of the BERT model.*

* How does one specify a task for the BERT model?
* The hugging face community

*Give some perspective into the effectiveness of this model, preferably in comparison to a self-trained neural network. Use GLUE/super-GLUE as the benchmark.*

* Some fancy pictures with explanation. (all big models are bert based)

*END OF THEORY: begin implementation part.*

* Summarize the theory by listing things that are needed in this thesis, which must be made.
* General input and output of the thesis NLP pipeline.
* Transformer based word analysation pipelines (different types of tasks POS tagging, NER, zero shot classification).
* Finetuning a BERT model.
  1. **Implementation of BERT models**

*Introduction to finetuning of BERT models*

*Finding the training data to finetune the BERT model.*

* Hugging face dataset community
* Treebanks
* Spacey datasets maybe?
  1. **Training/finetuning of BERT models Not sure about the specifics of this chapter and on yet.**

*Training structure introduction (which hyperparameters are tuned)*

*Weight initializations (probably only transfer learning)*

*ADAM optimizer (or others)*

*Learning rate tuning*

*Regularization techniques (dropout, weight regularization, early stop)*

*Evaluation of finetuned BERT models*

*GLUE/SUPER-GLUE score of training.*

* 1. **Evaluation of self-trained model compared to finetuned model from hugging face.**

*Comparison of trained model with alternative finetuned model found on hugging face.*

*Beg for mercy, as to why I’ve only trained one BERT model, when I intend to use several fine-tuned hugging-face BERT models to analyse my text.*

1. **Segment 2 (PARSER)**
   1. **Introduction to the task (whats input, whats output)**
   2. **Example natural language deconstruction (we need a memory storage for key words, we need a classifier, we need an action sequence handler, connection to the Ur robot)**
   3. **Memory storage (.txt files with keywords, neural network with parameters, new word embedding maybe)**
   4. **Classifier type (non-neural network “skill” picking with parameters “Move\_to(destination)”, “change location(destination1, destination2)”**
   5. **Action sequence handler (Idea of handling skills in a sequential manner with priorities)**
   6. **Rtde connection from parser to URscripting**

**Segment 3 (UR robot kinematics/URScripting)**

1. **Introduction to the task (bit weird, what should output be “robot movement patterns”, and then what skills should be implemented)**
2. **Robot kinematics and what could be possible using skills.**

**Integration:**

**Evaluation:**

**Discussion:**

**Conclusion:**

**Literature:**